Notice d'utilisation du contact dans *Code_Aster*

Résumé :

Ce document décrit la démarche à suivre pour la prise en compte de conditions de contact-frottement dans les études non linéaires. Dans un premier temps, on rappelle ce que signifie prendre en compte du contact-frottement en mécanique des structures, puis on trace les grandes lignes d'un problème de contact dans *Code_Aster* : appariement et résolution.

La définition du contact s'effectue avec la commande DEFI_CONTACT tandis que la résolution se fait avec les commandes STAT_NON_LINE ou DYNA_NON_LINE. On formule des préconisations pour la paramétrisation de l'appariement et le choix des méthodes de résolution dans ces opérateurs.

Enfin diverses méthodologies sont évoquées (contact avec une surface rigide, récupérer une pression de contact en post-traitement, grandes déformations et contact, mouvements de corps rigide bloqués par le contact, …). Elles permettent de surmonter les difficultés fréquemment rencontrées dans les études. Dans cette section, sont aussi abordées les modélisations alternatives du phénomène de contact-frottement par des éléments de joints ou des éléments discrets (à travers la loi de comportement).

Table des Matières

1 Introduction

1.1 Objet de ce document

Dire que deux corps solides mis en contact ne s'interpénètrent pas mais qu'au contraire un effort réciproque est exercé de l'un sur l'autre et que cet effort disparaît lorsque les corps ne se touchent plus, relève du bon sens. C'est la définition la plus succincte que l'on puisse faire du problème de « contact » : pourtant faire respecter ces conditions dans un code de calcul des structures comme *Code_Aster* demande beaucoup d'efforts.

Résoudre le problème de contact, c'est finalement imposer une condition aux limites sur certains degrés de liberté de déplacement et trouver une inconnue supplémentaire, l'effort réciproque s'exerçant entre les deux corps.

La difficulté provient de la forte non-linéarité induite par cette « pseudo-condition aux limites ». En effet la condition à imposer aux déplacements (pour empêcher toute interpénétration) dépend elle même des déplacements (qui vont déterminer en quel point les surfaces entrent en contact).

La non-linéarité due à la prise en compte de contact est séparée dans *Code_Aster* en deux points :

- non-linéarité de contact(-frottement) : elle découle des conditions de contact(-frottement) qui ne sont pas univoques. On fait ici appel à un algorithme d'optimisation sous contraintes ou bien à une transformation du problème en une version non contrainte.
- non-linéarité géométrique : elle découle des grands glissements relatifs susceptibles de se produire entre les surfaces en contact (méconnaissance *a priori* des surfaces finales effectives de contact). On fait ici appel à un algorithme de point fixe ou de Newton couplé à une recherche géométrique.

Dans *Code_Aster*, en présence de contact, l'utilisateur doit *a minima* identifier les surfaces potentielles de contact. La technique de résolution repose ensuite sur deux étapes fondamentales :

- Phase d'appariement : elle permet de traiter la non-linéarité géométrique comme une suite de problèmes en petits glissements (où le problème est géométriquement linéaire). La technique pour déterminer les surfaces effectives de contact et des conseils de paramétrage de cette phase sont donnés à la section 2.
- Phase de résolution : elle permet de résoudre le problème d'optimisation sous contraintes lié à la non-linéarité de contact et éventuellement de frottement. Les différents algorithmes d'optimisation disponibles sont présentés dans la section 3. On y donne un cheminement pour choisir un algorithme adapté à son cas d'étude.

Il est indispensable d'avoir compris que le contact-frottement est une non-linéarité à part entière au même titre que les non-linéarités matériaux (loi de comportement non linéaire) et cinématiques (grands déplacements, grandes rotations). Elle demande donc à la fois de connaître les bases de la théorie du contact et de comprendre le traitement de celui-ci dans *Code_Aster* afin de faire les bons choix de modélisation (maillage et mise en données).
Ce document est là pour assister l'utilisateur dans ces choix.

1.2 Une question de vocabulaire

Afin de faciliter la lecture, on donne ici quelques-uns des termes abondamment utilisés dans ce document.
Lorsque l'on parle de mécanique du contact, on utilise deux grandeurs caractéristiques :

- le jeu souvent noté g ou d. Il caractérise la distance signée entre deux surfaces de contact ;
- la densité d'effort de contact p. Il s'agit de l'effort réciproque exercé par un solide sur l'autre lorsque le jeu est fermé (nul). Il est porté par la normale aux surfaces de contact. On utilisera aussi abusivement le terme de pression de contact.

Ces grandeurs interviennent dans les conditions de Hertz-Signorini-Moreau de respect du contact (*cf.* [R5.03.50]).

En présence de frottement, on introduit par ailleurs :
- la direction de glissement $\vec{t}$
- la densité d'effort de frottement τ, portée par $-\vec{t}$.

Dans *Code_Aster*, on utilise un critère de frottement de Coulomb, les conditions de frottement sont décrites dans [R5.03.50].

1.3 Modélisations alternatives du contact-frottement

Si la manière de traiter le phénomène de contact-frottement décrite en introduction et dans l'essentiel de ce document est la plus répandue, elle n'est pas la seule. *Code_Aster* propose ainsi deux modélisations alternatives des interactions mécaniques :
- éléments de joints (hydro-)mécaniques (modélisations `*_JOINT*`) pour la représentation de l'ouverture d'une fissure sous la pression d'un fluide et du frottement entre les parois de la fissure fermée
- éléments discrets de choc (modélisations `*_DIS_T*`) pour la représentation d'un contact ponctuel par des ressorts avec prise en compte éventuelle du frottement

Ces deux autres modélisations s'appuient toutes deux sur des éléments finis et donc sur des lois de comportement spécifiques (`JOINT_MECA_FROT` pour les éléments de joints et `DIS_CHOC` pour les éléments discrets).

Plus de précisions sur ces éléments sont fournies aux §4.8 et §4.9.

Pour finir, on notera qu'il est possible de modéliser du contact sur les bords d'une fissure représentée avec la méthode X-FEM. On se reportera à la notice [U2.05.02] pour plus d'informations.

2 Appariement

2.1 Notion de zones et de surfaces de contact

C'est toujours à l'utilisateur de définir les surfaces **potentielles** de contact : il n'existe pas dans *Code_Aster* de mécanisme automatique de détection des interpénétrations possibles dans une structure.

L'utilisateur fournit donc dans le fichier de commandes une liste de couples de surfaces de contact. Chaque couple contient une **surface** dite « maître » et une **surface** dite « esclave ». On appelle « **zone** de contact » un tel couple.
Les conditions de contact seront imposées zone par zone. Faire respecter le contact consiste à **empêcher les nœuds esclaves de pénétrer à l'intérieur des surfaces maîtres** (en revanche l'inverse est possible).
Sur l'exemple ci-dessous (*cf.* Figure 2.1-1), la structure étudiée est constituée de trois solides, on a défini trois zones de contact potentielles symbolisées par les ellipses rouges. Comme leur nom l'indique ces zones de contact déterminent des parties de la structure où des corps sont **susceptibles** d'entrer en contact. Cela signifie que l'on y fait respecter les conditions de contact-frottement, l'activation effective de contact dépendant *in fine* du chargement imposé.

Il n'y a aucune restriction sur le nombre de zones de contact. Les zones doivent cependant être séparées, c'est-à-dire que l'intersection de deux zones distinctes doit être vide[1]. Par ailleurs, au sein d'une zone, les surfaces maîtres et esclaves d'une même zone doivent aussi avoir une intersection nulle : si ce n'est pas le cas, le calcul est arrêté. Lorsqu'un nœud est obligatoirement commun aux surfaces maîtres et esclaves, à cause d'une contrainte de maillage par exemple, se reporter au §2.3.4 pour une solution. Dans le cas où une formulation continue est utilisée (*cf.* 3.1.3), les surfaces esclaves doivent impérativement être deux à deux disjointes.

1 Plus précisément c'est l'intersection des surfaces esclaves qui doit être vide

Il ne faut pas hésiter à décrire de larges zones de contact pour éviter toute interpénétration. C'est le nombre de nœuds de la surface esclave qui est déterminante dans le coût de calcul. La surface maître peut, elle, être aussi grande qu'on le souhaite.

Il est impératif que les nœuds des surfaces de contact (maîtres et esclaves) portent tous des degrés de liberté de déplacement (`DX`, `DY` et éventuellement `DZ`), c'est-à-dire qu'ils appartiennent à des mailles du modèle. Un message d'erreur arrête l'utilisateur si ce n'est pas le cas. On se reportera au §4.4 pour la modélisation d'un contact avec une surface rigide.

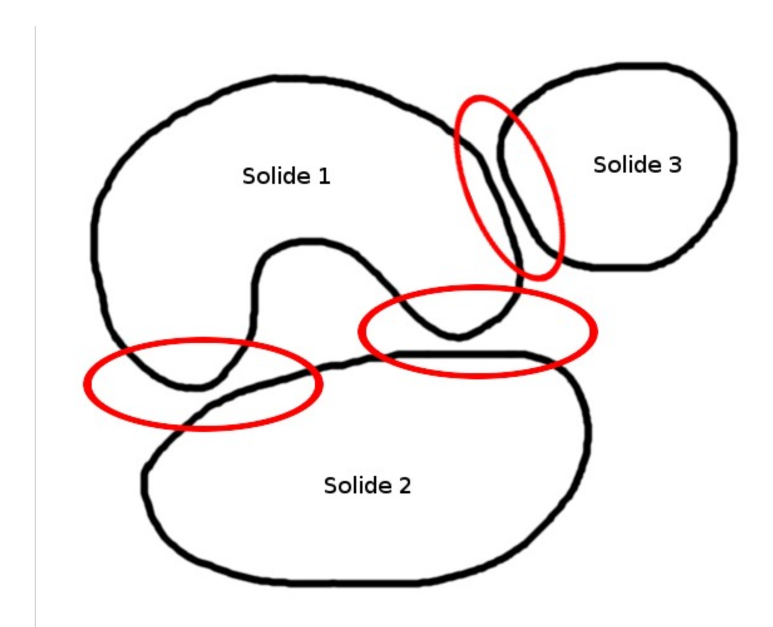


Figure 2.1-1: Définition de trois zones de contact

2.2 Choix des surfaces maîtres et esclaves

Comme on vient de le dire, chaque zone de contact est constituée d'une surface maître et d'une surface esclave. Dans l'état actuel, on ne peut pas faire d'auto-contact dans *Code_Aster* (sauf dans les rares cas où l'on peut prédire la future zone de contact et définir ainsi un esclave et un maître).

Le besoin de différencier les deux surfaces provient de la technique adoptée dans le calcul du jeu. Ce calcul est réalisé dans une phase que l'on nomme **appariement**. Le jeu est défini en tout point de la surface esclave (pour les méthodes discrètes il s'agit des nœuds, pour les méthodes continues des points d'intégration) comme la distance minimale à la surface maître. Cette dissymétrie implique un choix qui peut *a priori* s'avérer difficile (comment décider ?). Les points qui doivent prévaloir dans ce choix sont donnés dans les paragraphes suivants.

On renseigne ces surfaces dans l'opérateur `DEFI_CONTACT` sous le mot-clé facteur `ZONE`.

2.2.1 Cas où une surface doit être choisie comme maîtresse (`GROUP_MA_MAIT`)

Lorsqu'une de ces conditions est réunie :
- une des deux surfaces est **rigide** (a) ;
- une des deux surfaces **recouvre** l'autre (b) ;
- une des deux surfaces a une rigidité apparente **grande** devant l'autre (« apparente » au sens où l'on ne parle pas des modules d'Young mais des raideurs en $N.m^{-1}$) (c) ;
- une des deux surfaces est maillée beaucoup plus **grossièrement** que l'autre (d) ;

alors celle-ci doit être choisie comme surface maître.

2.2.2 Cas où une surface doit être choisie comme esclave (GROUP_MA_ESCL)

Lorsqu'une de ces conditions est réunie :
- une des deux surfaces est **courbe** (a) ;
- une des deux surfaces est plus **petite** que l'autre (b) ;
- une des deux surfaces a une rigidité apparente **petite** devant l'autre (c) ;
- une des deux surfaces est maillée beaucoup plus **finement** que l'autre (d) ;

alors celle-ci doit être choisie comme surface esclave.

2.2.3 Cas général

Lors de l'étude de structures complexes, il arrive que les règles données aux §2.2.1 et §2.2.2 soient difficiles à appliquer. Par exemple lorsqu'un solide est presque rigide (vis-à-vis de l'autre solide) et qu'il est courbe, la règle (a) ne permet pas de décider : faut-il privilégier le caractère courbe ou bien le caractère rigide ?
Dans ces situations « l'art de l'ingénieur » doit prévaloir. Dans notre exemple, si les deux solides subissent de faibles glissements, le caractère courbe du solide rigide n'aura que peu d'influence et on choisira donc ce dernier comme surface maître.

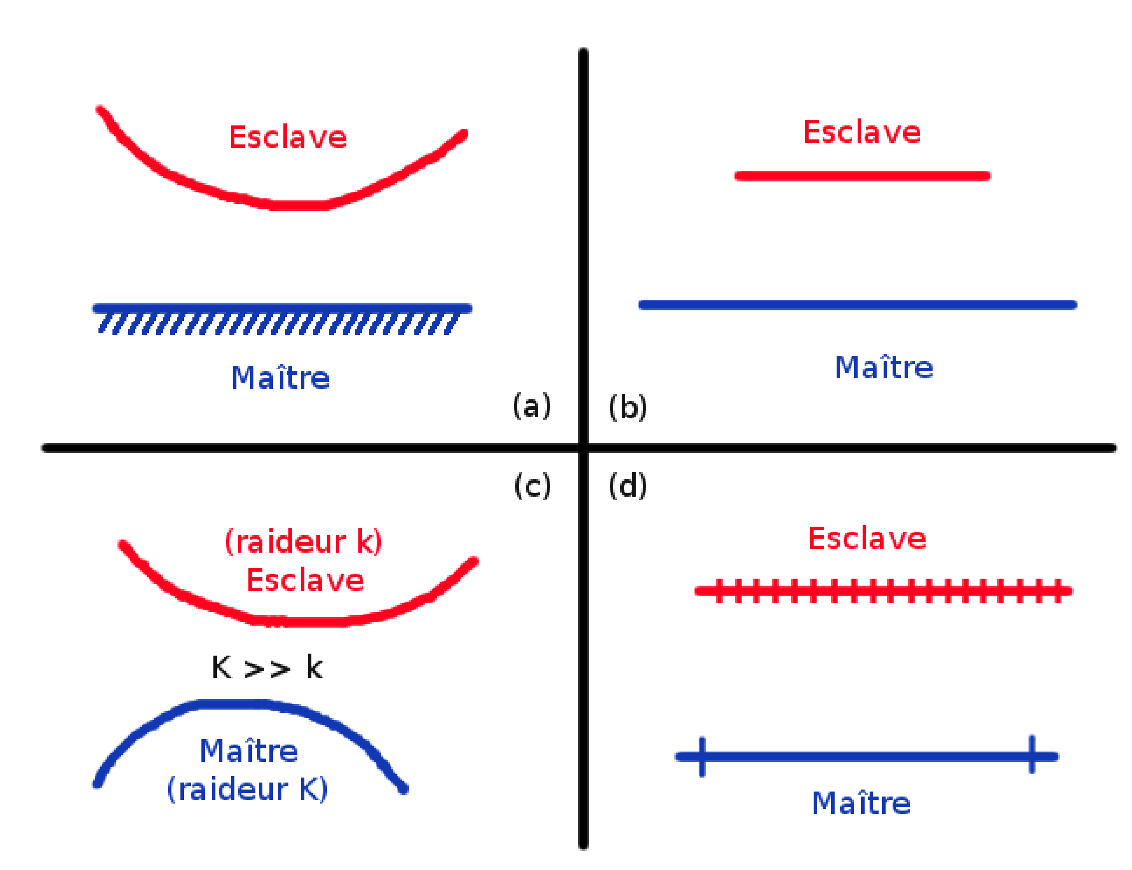


Figure 2.2.3-1: Choix de surfaces maître et esclaves suivant différentes situations

2.2.4 Orientation des normales

Il est primordial de toujours orienter les **normales** des surfaces de contact de manière à ce qu'elles soient **sortantes**. On peut le faire à l'aide de l'opérateur MODI_MAILLAGE. Selon que la surface à orienter est une maille de peau d'un élément massif, une coque ou bien une poutre, on utilisera respectivement les mot-clés ORIE_PEAU_2D ou ORIE_PEAU_3D, ORIE_NORM_COQUE, ORIE_LIGNE. Dans le cas de ORIE_LIGNE, on oriente la tangente, de sorte à pouvoir produire systématiquement la normale par un produit vectoriel.

Par défaut (mot-clé VERI_NORM de DEFI_CONTACT), la bonne orientation des normales est vérifiée et on arrête l'utilisateur si besoin.

2.2.5 Finesse et degré de maillage des surfaces courbes

Lorsque les surfaces de contact sont courbes, il faut garantir la bonne continuité de la normale aux facettes. Pour cela, on peut soit :
- mailler finement en linéaire et utiliser l'option de lissage (*cf.* §2.3.2)
- mailler en quadratique

Pour que le maillage quadratique conserve son intérêt, il faut avoir placé les **nœuds milieux sur la géométrie** dans le mailleur et non pas avoir utilisé l'opérateur CREA_MAILLAGE/LINE_QUAD de *Code_Aster*.

Dans le cas de surfaces de contact quadratiques, **en formulation discrète il ne faut pas que les surfaces de contact soient constituées de mailles quadrangulaires à 8 nœuds** (QUAD8) et on préférera donc plutôt les mailles à 9 nœuds (QUAD9). On transformera alors les HEXA20 en HEXA27 et les PENTA15 en PENTA18 (avec l'opérateur CREA_MAILLAGE). À l'heure actuelle, les maillages mixtes constitués à la fois d'HEXA20 et de PENTA15 ne sont pas transformables par CREA_MAILLAGE.
En formulation continue, pour des mailles de bords courbes, l'utilisation d'éléments QUAD8 ou TRIA6 peut entraîner des violations de la loi de contact : ce dernier est vérifié en moyenne. On observe alors des jeux légèrement positifs ou légèrement négatifs en présence de contact, ce qui peut perturber les résultats proches de la zone de contact ou les calculs en reprise avec état initial.

Lorsque à la fin d'un calcul on remarque un fort taux d'interpénétration des nœuds maîtres à l'intérieur des surfaces esclaves (ce qui est possible contrairement à l'inverse), cela signifie généralement que le maillage d'une ou des deux surfaces est trop grossier ou qu'il y a une trop grande différence de finesse entre les deux maillages des surfaces. On peut alors soit raffiner, soit inverser maître et esclave.

Si une surface est rigide (et donc maître), un maillage grossier est suffisant sauf bien sûr dans les zones courbes.

Enfin dans le cas particulier d'un **contact cylindre-cylindre** ou **sphère-sphère**, il faut veiller à **mailler suffisamment chaque surface** pour éviter de laisser trop de vide entre elles. En effet dans *Code_Aster*, on ne fait pour l'instant pas de repositionnement de nœuds ni de projections sur des splines passant par la surface maître, un maillage trop grossier provoquera alors une **forte oscillation de la pression de contact** (détection du contact un nœud sur deux).

2.2.6 Angles vifs

Les algorithmes d'appariement fonctionnent moins bien en présence d'angles vifs, c'est pourquoi on évitera le plus possible d'en avoir dans le maillage des surfaces maîtres et esclaves. Par exemple on préférera modéliser un congé plutôt qu'un angle vif.

Si un angle vif est indispensable, on choisira la surface qui le porte comme esclave.

2.2.7 Qualité du maillage

La qualité des éléments surfaciques qui constituent la surface de contact maître a un impact direct sur la qualité de l'appariement. En effet des mailles distordues, par exemple, peuvent nuire à la précision des projections malgré la robustesse de l'algorithme : l'unicité de la projection n'est plus garantie.
Pour ces raisons, il est recommandé de vérifier la qualité des maillages produits et si nécessaire de corriger leurs défauts. Dans *Code_Aster*, la commande MACR_INFO_MAIL permet d'afficher la répartition des éléments en fonction de leur qualité.

2.3 Contrôle de l'appariement

2.3.1 Choix du type d'appariement

Dans *Code_Aster*, deux types d'appariement sont disponibles :

- « maître-esclave » (par défaut) : c'est le plus générique, il permet d'empêcher les nœuds de la surface esclave de pénétrer les mailles de la surface maître à l'aide de projections orthogonales (d'un nœud sur une maille).
- « nodal » : il permet d'empêcher les nœuds esclaves de pénétrer les nœuds maîtres suivant une direction (donnée par la normale esclave). C'est un appariement réservé aux maillages compatibles des surfaces de contact pour des calculs en petits glissements. Il n'est pas disponible en formulation continue (*cf.* §3.1.3).

On conseille de toujours choisir l'appariement « **maître-esclave** », la valeur par défaut.

2.3.2 Lissage des normales

Comme son nom l'indique cette option permet de lisser les normales. Elle est particulièrement utile dans le cas de surfaces courbes maillées en linéaire. Ce procédé est fondé sur la moyenne des normales aux nœuds, puis leur interpolation à partir des fonctions de forme et des normales moyennées, il permet d'assurer la **continuité** de la normale aux nœuds.

La normale n'est alors plus la normale géométrique, on prendra donc la précaution (conseillée de toute manière) de bien vérifier visuellement les résultats.

Une vérification de la facettisation des surfaces est effectuée automatiquement en fin de pas de temps. Elle émet un message d'information lorsque celle-ci devient trop importante et il est alors conseillé d'activer le lissage.

2.3.3 Choix des normales

On conseille de toujours de laisser les valeurs par défaut : `NORMALE='MAIT'`, `VECT_MAIT='AUTO'`. C'est-à-dire que la relation de non-pénétration est écrite à partir de la normale maître, déterminée grâce au maillage.

Cependant il existe quelques rares situations où l'on peut vouloir imposer le choix de la normale : il s'agit essentiellement du traitement du contact poutre-poutre (en 2D uniquement) et du cas où la surface maître est une maille de type `POI1`. On renvoie au §3.1.6 de [U4.44.11] pour plus de détails.

2.3.4 Exclusion de nœuds esclaves de l'appariement

Le mot-clé `SANS_GROUP_NO`/`SANS_NOEUD` sert à exclure de l'appariement des nœuds esclaves. Il peut y avoir plusieurs raisons à cela :

- surface maître et esclave ont une intersection non vide (fond de fissure, blocage de mouvements de corps rigide) ; les nœuds communs n'ont pas besoin d'être traités par le contact, ils doivent donc être exclus.
- il existe déjà sur les nœuds esclaves considérés des relations linéaires (conditions aux limites, blocage de mouvements de corps rigide) ; si celles-ci interfèrent avec la direction du contact (respectivement du frottement), on conseille en général de privilégier les conditions aux limites et donc ne pas résoudre le contact sur ces nœuds.

Une erreur fatale est émise lorsqu'il existe des nœuds communs aux surfaces maîtres et esclaves et que ces derniers n'ont pas été exclus.

2.4 Comprendre la non-linéarité géométrique

Comme on l'a expliqué, la non-linéarité géométrique découle du fait que l'on doit appliquer des conditions de contact-frottement sur une configuration géométrique que l'on ne connaît pas. Dans cette section, on fait une petite digression afin d'expliquer la démarche adoptée pour surmonter cette difficulté.

2.4.1 Hypothèse de petits glissements

La phase d'appariement est une phase préalable à la formulation des conditions de contact à résoudre. En pratique cela signifie :

- pour les méthodes discrètes, la construction d'une matrice A (pour Appariement) telle que multipliée par l'incrément de déplacement δu depuis la configuration appariée, elle donne l'incrément de jeu (linéarisé).
- pour la méthode continue, l'association entre un point de contact et son projeté dans l'espace paramétrique de la maille maître appariée. C'est en actualisant les coordonnées de la maille maître avec le déplacement δu que l'obtient les nouvelles coordonnées (linéarisées) du projeté.

De même que les conditions d'équilibre, **les conditions de contact s'expriment sur la configuration déformée** (ou finale). Cette configuration n'est pas connue *a priori*.
L'hypothèse de faibles glissements relatifs des surfaces en contact est l'analogue de l'hypothèse de petites perturbations (pour l'écriture des relations d'équilibre).

Elle consiste à dire que la configuration finale des surfaces en contact est peu différente de la configuration initiale, ce qui permet donc de réaliser l'appariement une fois pour toutes en début de calcul sur la configuration initiale. Puis d'utiliser les conditions établies sur cette configuration pour tout le calcul.

Un tel problème est alors linéaire géométriquement : seule la non-linéarité de contact-frottement demeure, elle est traitée avec des algorithmes adaptés (*cf.* section 3).

2.4.2 Cas général

Pour traiter des problèmes de grands glissements relatifs des surfaces en contact, deux possibilités existent : l'utilisation d'une boucle de point fixe pour se ramener aux cas de petits glissements ou bien pour la formulation continue (§3.1.3) la résolution simultanée au sein de l'algorithme de Newton.

2.4.2.1 Boucle de point fixe (ALGO_RESO_GEOM='POINT_FIXE')

La démarche adoptée est très similaire à la résolution d'un problème non-linéaire par la méthode de Newton. **On transforme un problème non-linéaire géométrique en une suite de problèmes linéaires géométriques**. Pour cela on va résoudre une suite de problèmes dans l'hypothèse de petits glissements.
C'est-à-dire que l'on réalise un appariement (sur une configuration initiale équilibrée) et une résolution de Newton (avec résolution du contact comme on l'expliquera dans la section 3). Ceci nous donne une nouvelle configuration ; si cette configuration est « proche » de la configuration initiale alors on a convergé (c'était donc la configuration finale), sinon on boucle : on refait un appariement puis une résolution … et ainsi de suite jusqu'à trouver la configuration finale (*cf.* Figure 2.4.3.2-1).

La difficulté est dans la caractérisation de la convergence de ce processus de point fixe. Qu'est-ce que deux configurations « proches » ? Dans *Code_Aster*, il s'agit de deux configurations dont le vecteur déplacement « mécanique » pour passer de l'une à l'autre (c'est-à-dire l'incrément de déplacement obtenu par Newton restreint aux degrés de liberté DX, DY, DZ) a une norme infinie petite devant la norme infinie du vecteur déplacement précédent.

Cela implique que l'on fait donc toujours au moins deux itérations de géométrie avec ce critère (afin de se donner un vecteur déplacement initial). On renvoie au paragraphe 3.7 de [R5.03.50] pour l'expression exacte de la norme infinie.

2.4.2.2 Algorithme de Newton généralisé (ALGO_RESO_GEOM='NEWTON')

La formulation continue (§3.1.3) offre la possibilité de traiter la non-linéarité géométrique directement au sein de l'algorithme de Newton. Pour cela un appariement est réalisé à chaque itération et les termes géométriques de la matrice tangente sont eux aussi réactualisés.

Le critère de convergence géométrique devient ainsi un critère supplémentaire de l'algorithme de Newton : l'incrément de déplacement doit tendre vers zéro.

2.4.3 Convergence de la boucle de géométrie

2.4.3.1 Linéarisation de la normale

L'appariement fournit deux informations :
le jeu sur la configuration appariée,
les coordonnées du point de contact potentiel sur la surface maître.

Que l'on utilise une méthode de contact discrète ou continue, on doit dériver (linéariser) le principe des puissances virtuelles, en particulier les termes de contact.

L'hypothèse de point fixe conduit à faire également l'hypothèse d'une faible variation de la normale au cours des itérations de Newton (d'un pas de temps). Cette hypothèse est cohérente avec l'hypothèse de petits glissements. L'opérateur « jeu » est ainsi linéarisé plus facilement. Il reste en particulier constant au cours d'une itération de géométrie. Ceci a aussi une autre implication : la convergence géométrique vers la configuration finale en mode point fixe peut parfois être très lente.
À l'inverse, la résolution par une méthode de Newton généralisée, possible uniquement en formulation continue (§3.1.3), présente une convergence beaucoup plus rapide mais peut s'avérer moins robuste. C'est pourquoi ce n'est pas le réglage par défaut. Pour l'activer, on utilisera le mot-clé `ALGO_RESO_GEOM`.

2.4.3.2 Critère de convergence géométrique

On a vu plus haut que la convergence de la boucle de géométrie se fait sur un critère géométrique : la différence des vecteurs déplacements entre deux configurations géométriques successives est petite en relatif. Pour le cas particulier de l'algorithme de Newton généralisé en formulation continue, le critère s'applique directement à l'incrément de déplacement de Newton.

Pour la résolution par boucle de point fixe, « petit » correspond par défaut à inférieur à 1% du déplacement depuis le début du pas de temps tandis qu'avec l'algorithme de Newton généralisé, il s'agit de 0,0001 % (celui-ci bénéficiant dans les cas favorables d'une meilleure convergence, il est possible d'exiger un critère plus dur).

Lorsque à la suite d'un calcul, on observe une interpénétration des nœuds esclaves dans la surface maître, la seule explication[2] est une non-vérification du critère géométrique.
Il ne faut alors pas hésiter à durcir le critère. Pour cela on utilise le mot-clé `RESI_GEOM=0.005` de sorte à abaisser le seuil autour de 0,5% par exemple.

Parfois la convergence géométrique est simplement lente, dans ce cas il suffit d'augmenter le nombre maximum d'itérations tolérées : `ITER_GEOM_MAXI=20` par exemple.

Dans certaines situations, on a beau jouer sur la valeur du critère ou le nombre d'itérations, le calcul ne converge pas : il cycle. Plusieurs possibilités s'offrent alors à l'utilisateur :

- l'explication la plus courante est une mauvaise discrétisation des surfaces de contact (c'est-à-dire un maillage trop grossier, une différence de finesse entre les deux surfaces ou bien un mauvais choix des surfaces maîtres et esclaves). On renvoie alors au §2.2.
- lorsque les surfaces de contact sont courbes et maillées relativement grossièrement, l'explication peut venir d'une trop grande discontinuité de la normale (facettisation). **L'activation du lissage facilite alors très souvent la convergence** (*cf.* §2.3.2). Cela ne doit cependant pas empêcher l'utilisateur de revoir son maillage.
- si le cas est vraiment pathologique, il faut forcer soi-même le nombre de réactualisations en utilisant `REAC_GEOM='CONTROLE'` et `NB_ITER_GEOM=n`. Après n itérations de géométrie, le calcul passera au pas de temps suivant quelle que soit la valeur du critère géométrique mais il émettra une alarme lorsque le critère n'est pas vérifié à moins de 1% (pour $n \geqslant 2$).

2 En formulation continue, cela peut se produire avec des éléments quadratiques courbes, *cf.* §2.2.5

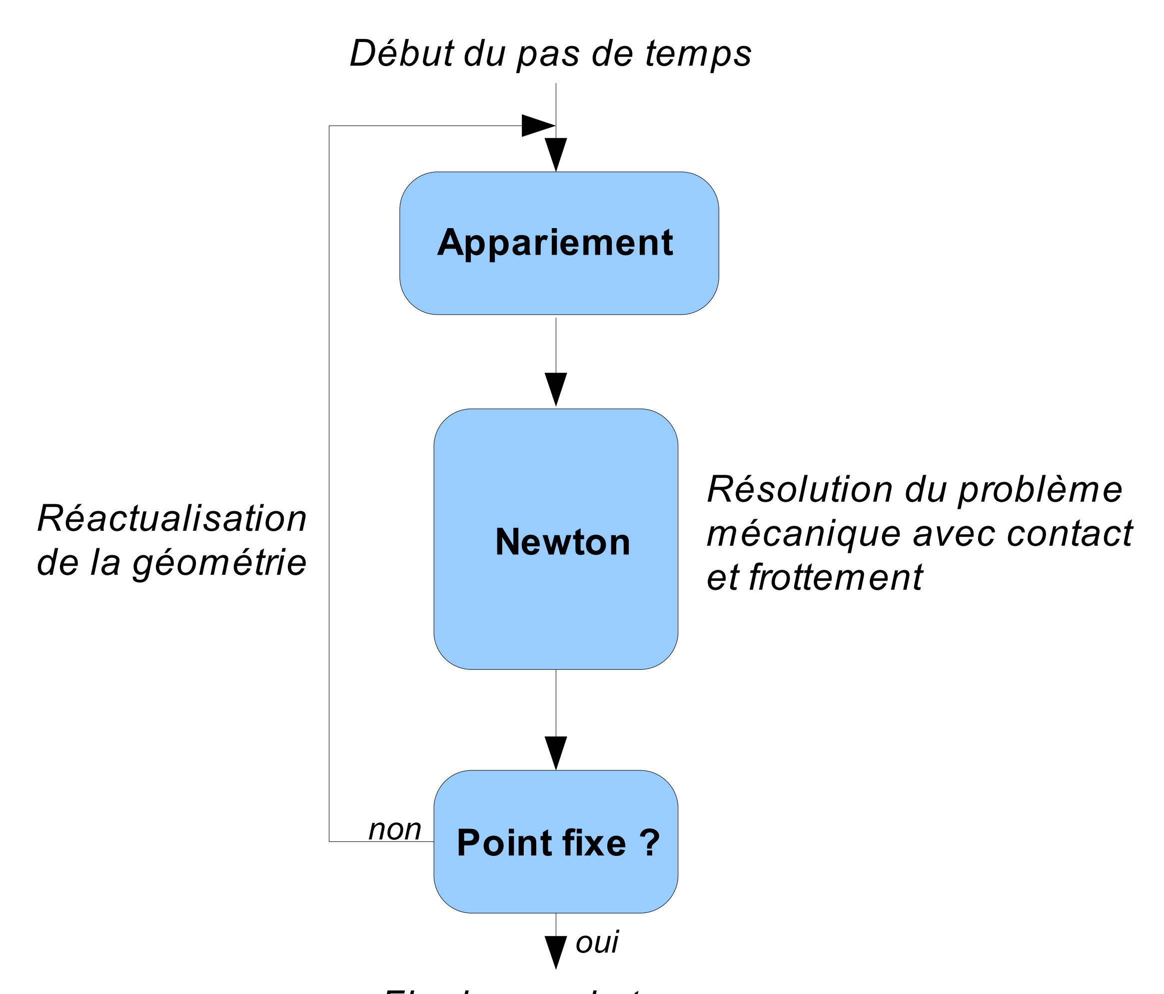


Figure 2.4.3.2-1: Traitement de la non-linéarité géométrique du contact par une boucle de point fixe

3 Résolution

3.1 Schéma général de l'algorithme de résolution

3.1.1 Définition

Ce que l'on appelle « résolution du contact », c'est l'opération consistant à résoudre le système formé par la juxtaposition des équations classiques de la mécanique et des équations de contact-frottement (l'aspect géométrique étant traité par l'appariement, il ne reste à ce stade que la non-linéarité de seuil du frottement et la non-linéarité de statut du contact).

Il faut noter que les deux formulations disponibles dans le code diffèrent notablement sur ce point. Sans entrer dans les détails, on explique brièvement ces différences pour la suite.

Si les formulations discrète et continue reviennent bien à résoudre le même problème physique, comme leur nom l'indique elles ne le formulent pas numériquement de la même manière :

- en méthode discrète, les conditions de contact-frottement sont appliquées sur le système discrétisé par la méthode des éléments finis. On ne modifie donc **que** la résolution du système linéaire obtenu par Newton $Ku=f$: on obtient alors un système linéaire sous contraintes (ces contraintes sont des inégalités linéaires). Les méthodes discrètes font donc appel à des algorithmes d'optimisation pour le résoudre.
- en méthode continue, on écrit une formulation variationnelle pour les équations de contact-frottement, elles sont donc discrétisées comme pour le principe des travaux virtuels. La démarche adoptée pour résoudre le système non-linéaire obtenu est un découplage des non-linéarités ou une linéarisation de Newton associés à un lagrangien augmenté : le système devient donc linéaire et non contraint, il n'y a donc pas d'algorithme spécifique à appliquer comme en méthode discrète mais plutôt un choix de paramètres judicieux (dans le contrôle des boucles et le Lagrangien augmenté).

3.1.2 Formulation discrète

Pour illustrer la définition du paragraphe précédent, on donne Figure 3.1.2-1 le schéma général de l'algorithme dans le cas d'une formulation discrète. Nous pouvons faire les remarques suivantes sur ce schéma :

- il ne représente qu'un seul pas de temps en supposant que l'on se place en petits glissements (on ne fait donc pas apparaître la boucle externe, comme dans la Figure 2.4.3.2-1, traitant la non-linéarité géométrique et décrite au §2.4) ;
- dans ce schéma, apparaissent les trois étapes classiques d'une itération de Newton : assemblage et résolution du système linéaire, intégration de la loi de comportement, analyse de la convergence ;
- la particularité de la formulation discrète du contact consiste en **l'adjonction** d'une étape supplémentaire entre la résolution du système linéaire (sans contact) et l'intégration de la loi de comportement. **On peut voir cette étape comme un post-traitement de la solution du système sans contact.**

L'étape supplémentaire que réalise la boite « contact discret » a pour but la construction puis la résolution du système augmenté par les conditions de contact et de frottement. Deux approches existent pour formuler les conditions discrètes de contact-frottement :

- écriture d'un lagrangien et dualisation des conditions de contact-frottement, on augmente alors artificiellement la taille du système global à résoudre et on utilise un algorithme d'optimisation pour satisfaire les contraintes inégalités. Cette approche est traitée aux §3.2.1 et §3.3.2.
- pénalisation (ou régularisation) des conditions de contact-frottement, on conserve la même taille pour le système global mais on enrichit la matrice, il n'y a pas d'algorithme spécifique, c'est l'algorithme de Newton qui assure la convergence. En revanche le contact n'est résolu qu'approximativement et l'utilisateur doit fournir un ou des paramètres pour contrôler l'algorithme. Cette approche est traitée aux §3.2.2 et §3.3.3.

Ce que produit la boite « contact discret » en sortie est un champ de déplacement vérifiant les conditions de contact-frottement ainsi que des réactions de contact-frottement. Ces réactions sont utilisées dans la vérification de l'équilibre.

La formulation discrète s'appuie donc sur la résolution d'un problème mécanique sans contact ce qui a une conséquence importante : **on ne peut pas traiter simplement le cas d'une structure où le contact comme le frottement participent directement aux blocages des mouvements de corps rigide** (*cf.* §4.2).

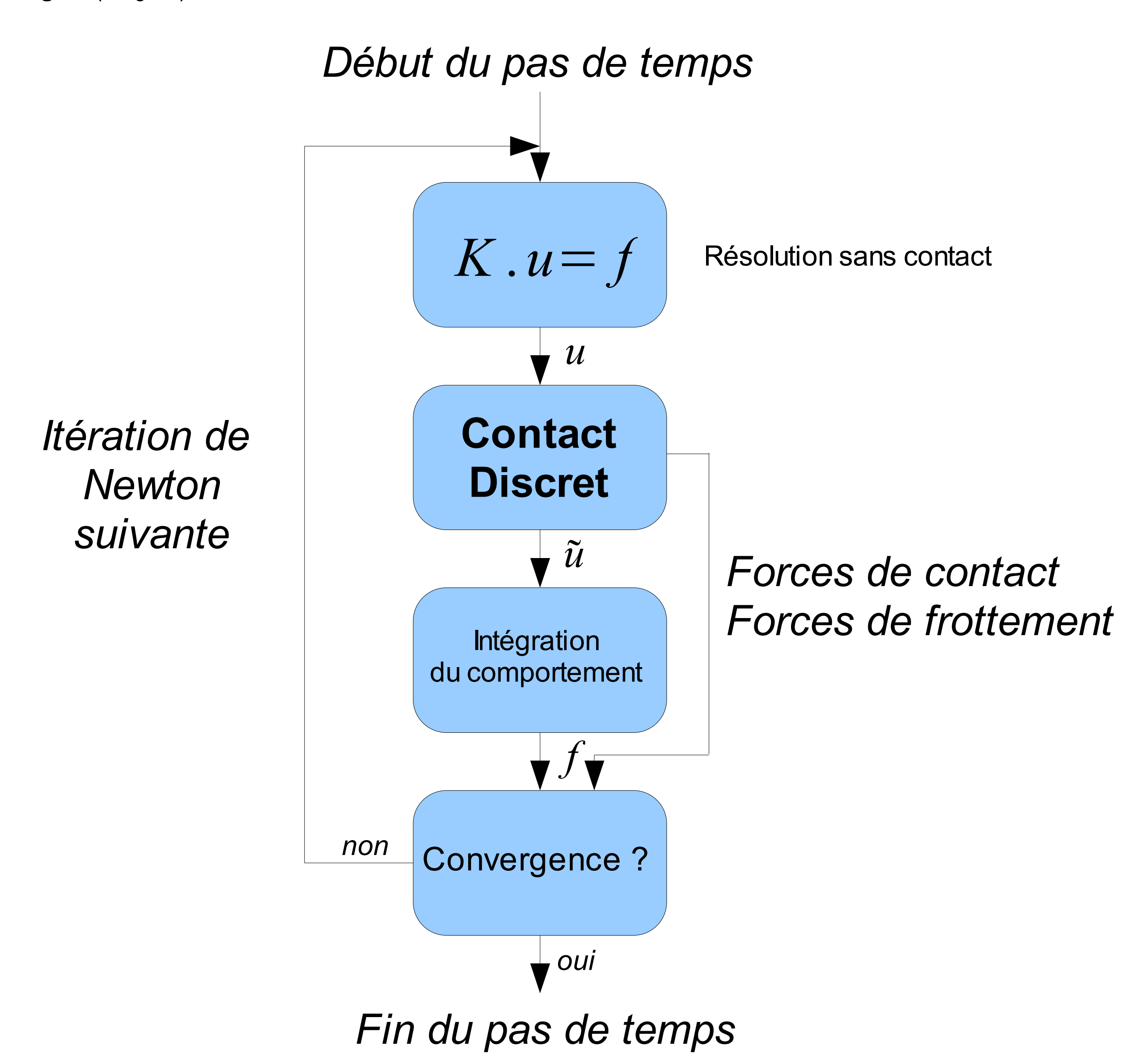


Figure 3.1.2-1: Algorithme général d'un pas de temps en formulation discrète (petits glissements)

3.1.3 Formulation continue

La Figure 3.1.3-1 donne l'algorithme général de résolution du contact-frottement avec une formulation continue, celui-ci diffère notablement du schéma en formulation discrète. Alors qu'avec cette dernière le contact-frottement est résolu par des sous-itérations (dans la boite « Contact Discret »), la formulation continue s'appuie sur un découplage des non-linéarités :

- la non-linéarité de frottement (le seuil de Coulomb dépend de la pression de contact qui est elle-même une inconnue) est traitée par un point fixe sur la valeur du multiplicateur de contact ou bien un algorithme de Newton généralisé

- la non-linéarité de contact s'appuie sur un algorithme des statuts (avec bascule par paquets) ou bien un algorithme de Newton généralisé

Lorsque toutes les non-linéarités sont découplées, il ne subsiste dans l'algorithme de Newton que les non-linéarités classiques matériaux et cinématiques.
Un Lagrangien augmenté permet de transformer le système contraint par les inégalités de contact-frottement en un système non contraint. Chaque itération de Newton en formulation continue ne coûte pas plus en mémoire que dans un calcul sans contact de taille équivalente contrairement à la formulation discrète. Néanmoins l'imbrication des boucles ou bien le traitement par l'algorithme de Newton généralisé impliquent un plus grand nombre d'itérations (de Newton).
En formulation continue, il existe des degrés de liberté supplémentaires dans la modélisation, conséquence de l'écriture variationnelle des conditions de contact, comme expliqué au §4.1.1.

Figure 3.1.3-1: Algorithme général d'un pas de temps en formulation continue avec point fixe (petits glissements)

3.2 Résolution d'un problème avec contact seul

3.2.1 Dualisation en formulation discrète (FORMULATION='DISCRETE')

3.2.1.1 Principe

La dualisation du système discret consiste en l'introduction d'un Lagrangien (cf [R5.03.50]). Le système à résoudre prend la forme suivante lorsqu'il est réduit sur les liaisons actives :

$$\begin{cases} \boldsymbol{C}.\delta\tilde{\boldsymbol{u}}+\boldsymbol{A}_{\mathrm{c}}^{T}.\boldsymbol{\mu}_i=\boldsymbol{F}_i \\ \boldsymbol{A}_{\mathrm{c}}.\delta\tilde{\boldsymbol{u}}=d_{i-1} \end{cases} \quad (1)$$

Sachant que la résolution du système sans contact a déjà été effectuée, on connaît la solution du système suivant :

$$\boldsymbol{C}.\delta\boldsymbol{u}=\boldsymbol{F}_i \quad (2)$$

La technique de résolution s'appuie alors sur l'utilisation du complément de Schur du système (1) pour transformer le système :

$$\boldsymbol{S}_{\mathrm{schur}}=-\boldsymbol{A}_c.\boldsymbol{C}^{-1}.(\boldsymbol{A}_c)^T \quad (3)$$

Le problème ainsi transformé a la taille du nombre de nœuds esclaves et il est plein. Deux algorithmes au choix sont disponibles pour traiter ce nouveau problème :

- une méthode de contraintes actives (ALGO_CONT='CONTRAINTE') s'appuyant sur la construction **explicite** et la factorisation du complément de Schur
- une méthode de gradient conjugué projeté (ALGO_CONT='GCP') s'appuyant sur la résolution **itérative** du système formé par le complément de Schur du système

Il est à noter que la dualisation requiert l'utilisation d'un solveur linéaire direct : dans *Code_Aster*, cela signifie 'MULT_FRONT' ou 'MUMPS'.
Chacun des 2 algorithmes cités ci-dessus réalise en effet des sous-itérations au cours desquelles il est nécessaire de résoudre le système linéaire (2) avec $\boldsymbol{C}$ la matrice de rigidité du système global sans contact (ce qui est beaucoup plus rapide si $\boldsymbol{C}$ est déjà factorisée).

3.2.1.2 Méthode 'CONTRAINTE'

S'appuyant sur une factorisation (donc un solveur direct) pour résoudre le système associé au complément de Schur, la méthode 'CONTRAINTE' **ne demande aucun paramétrage**. Par ailleurs sa convergence[3] est démontrée, ce qui explique qu'elle est la méthode par défaut en présence de contact.

Néanmoins l'utilisation d'un solveur direct présente un inconvénient majeur : **cet algorithme n'est pas adapté dès que le nombre de nœuds esclaves dépasse quelques centaines (500)**. En effet la factorisation d'une matrice pleine devient très vite rédhibitoire.

La construction du complément de Schur peut être accélérée en utilisant le paramètre NB_RESOL (*cf.* [U4.44.11], valeur par défaut 10) au détriment de la mémoire consommée (plus le nombre de degrés de liberté total est grand, plus l'augmentation de ce paramètre coûte cher). Afin d'optimiser un calcul avec la méthode des contraintes actives, il est conseillé de faire un calcul sur un pas de temps afin de trouver un compromis temps/mémoire (*cf.* [U1.03.03] pour la lecture des informations sur la mémoire consommée).

3 On utilise bien un solveur direct pour construire le complément de Schur mais la méthode des contraintes actives consiste à activer ou désactiver une par une les liaisons de contact jusqu'à satisfaire le système global, c'est donc un algorithme itératif.

3.2.1.3 Méthode 'GCP'

Lorsque que l'on ne peut plus utiliser la méthode de contact par défaut parce qu'elle coûte trop cher, une alternative est l'utilisation de la méthode 'GCP'. Comme on l'a dit plus haut cette méthode consiste en l'application d'un solveur itératif (gradient conjugué projeté) pour résoudre le problème dual.
Le principal avantage d'une telle méthode est de n'être plus limitée en taille de problème (plusieurs milliers de nœuds esclaves sont parfaitement atteignables). La contrepartie, propre à tout solveur itératif, est un paramétrage obligatoire pour l'utilisateur.
Cette méthode est utilisable en calcul parallèle, c'est d'ailleurs la seule méthode discrète à vraiment en profiter.

Comme tout solveur itératif, la méthode 'GCP' utilise un critère de convergence : il s'agit d'un critère sur la valeur du jeu. Donné par le mot-clé RESI_ABSO, il contrôle l'interpénétration maximum tolérée.
Il est obligatoire et s'exprime dans la même unité que celle utilisée pour le maillage. On conseille d'utiliser dans un premier temps un critère égal à 10^{-3} fois l'interpénétration moyenne quand le contact n'est pas pris en compte (cf. §4.6).

Si on constate des difficultés de convergence de l'algorithme du gradient conjugué projeté, il existe 2 paramètres sur lesquels, on conseille de jouer (de manière additive, c'est-à-dire l'un puis l'autre) :
- utiliser une recherche linéaire non-admissible (RECH_LINEAIRE='NON_ADMISSIBLE')
- utiliser un pré-conditionneur de Dirichlet (PRE_COND='DIRICHLET')

Le pré-conditionneur présente l'avantage d'être optimal et diminue donc sensiblement le nombre d'itérations nécessaires à la convergence. De plus lorsque l'on est proche de la solution, il permet de faire diminuer très rapidement le résidu et donc d'atteindre des critères d'interpénétrations très faibles.
Son inconvénient est un coût important qui peut souvent empêcher un gain en temps de calcul malgré la diminution du nombre d'itérations.
Pour cette raison, il est possible de ne demander son activation que lorsque le résidu a suffisamment diminué : le pré-conditionneur permet alors idéalement de converger en quelques itérations. La difficulté réside dans la quantification du « suffisamment diminué » ou autrement dit du voisinage de la solution. On contrôle ce déclenchement par le mot-clé COEF_RESI qui est le coefficient (inférieur à 1) par lequel il faut avoir multiplié le résidu initial (l'interpénétration maximale initiale donc) avant d'appliquer le pré-conditionneur. Un exemple de mise en œuvre de ce paramètre est donné dans le cas-test SSNA102E.

3.2.2 Pénalisation en formulation discrète : algorithme 'PENALISATION'

La pénalisation consiste à régulariser le problème de contact : au lieu de chercher à résoudre exactement les conditions sur le jeu et la pression, on introduit une relation approchée univoque qui implique que **l'on observera toujours une interpénétration lorsque le contact est établi**.

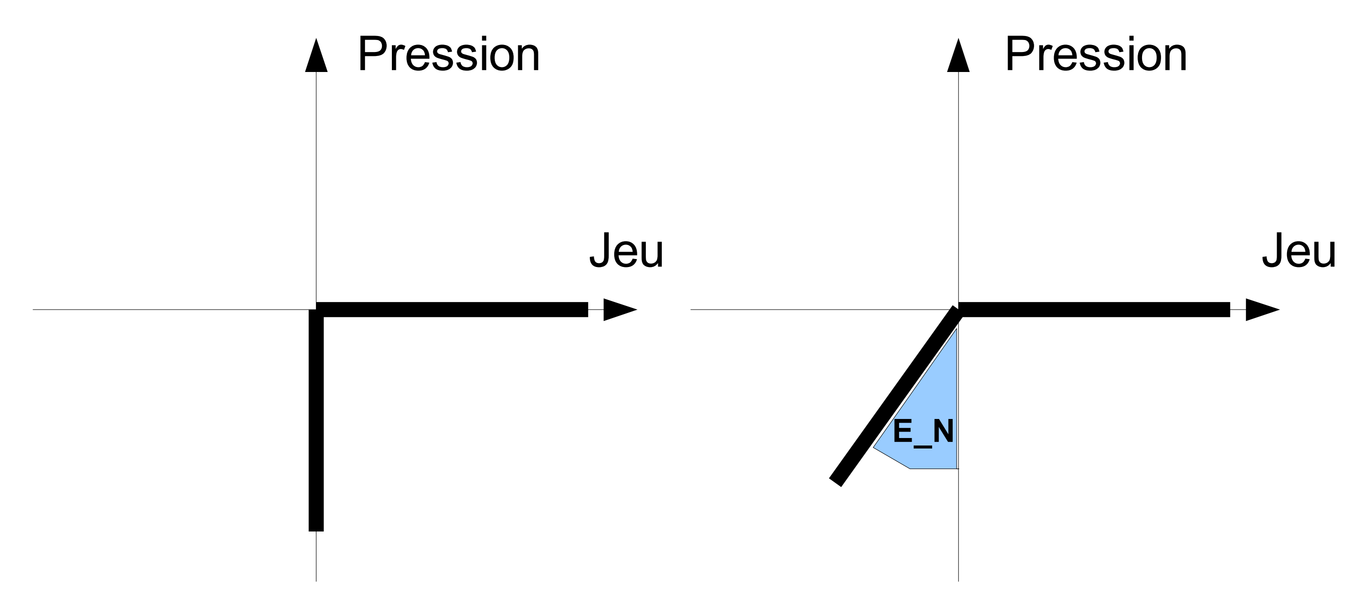


Figure 3.2.2-1: Condition de contact (à gauche) et régularisation (à droite)

Comme le montre la Figure 3.2.2-1 on ajoute un paramètre `E_N` pour régulariser la condition de contact : plus il est grand, plus on tend vers la condition exacte, plus il est petit, plus on tolère de l'interpénétration.
En formulation discrète, la notion de pression de contact n'existe pas car on raisonne sur les nœuds du maillage élément fini : on travaille donc avec des forces nodales (*cf.* §4.1). Le coefficient `E_N` dit de pénalisation a donc la dimension d'une raideur ($N.m^{-1}$).

On fait généralement l'analogie entre le coefficient de pénalisation et la raideur de ressorts unilatéraux que l'on placerait entre surface maître et esclave là où on observe de l'interpénétration.

On choisit généralement `E_N` par essais successifs :
- tout d'abord on commencera par prendre une valeur égale à 10 fois le plus grand module d'Young de la structure multiplié par une longueur caractéristique de celle-ci ;
- si le calcul donne un résultat (satisfaisant ou pas), on augmentera ensuite la valeur en la multipliant par 10 à chaque fois jusqu'à obtenir un résultat stable en termes de déplacements et surtout en termes de contraintes.

L'avantage de la méthode de pénalisation est **de ne pas augmenter la taille du système contrairement à la dualisation, mais aussi de ne pas restreindre le choix du solveur linéaire**. La contrepartie est une sensibilité au coefficient de pénalisation qui implique de mener systématiquement une étude paramétrique avant de se lancer dans de longs calculs (*cf.* [U1.04.00] et [U2.08.07] pour le lancement de calculs paramétriques distribués).

Pour aider à calibrer le coefficient de pénalisation, il existe un mécanisme d'adaptation automatique s'appuyant sur la commande `DEFI_LIST_INST` [U4.34.03]. On trouvera un exemple de mise en œuvre dans le cas-test SDNV103I [V5.03.103].

3.2.3 Formulation `'CONTINUE'`

Pour le problème de contact seul, la méthode continue présente l'avantage comme la méthode (discrète) des contraintes actives de ne demander aucun réglage par l'utilisateur.

Comme elle **n'est pas dépendante d'un solveur direct**, il est possible d'utiliser un solveur itératif (comme `'GCPC'` ou `'PETSC'`) associé aux pré-conditionneurs `'LDLT_SP'` ou `'LDLT_INC'` pour gagner énormément sur le temps de calcul. Dans ce cas, il est conseillé d'activer la méthode de Newton-Krylov (*cf.* mot-clé `METHODE` de `STAT_NON_LINE` [U4.51.03]) qui permet d'adapter automatiquement le critère de convergence du solveur itératif.

Le principal avantage de la méthode continue est de proposer via le degré de liberté LAGS_C (dans le champ DEPL) l'**accès à la pression de contact sur la surface esclave**.
On attire cependant l'attention sur le fait que cette quantité n'est en fait qu'**une densité de force de contact par unité de surface exprimée sur la configuration de référence**. En particulier, en grandes déformations, on ne peut plus la qualifier de pression car elle n'a plus de sens physique.

En formulation continue deux algorithmes existent :

- méthode de point fixe sur les statuts de contact : l'état des statuts de contact est évalué dans une boucle externe à la boucle de Newton
- méthode de Newton généralisée : les statuts de contact sont évalués à chaque itération de Newton (c'est le défaut)

Pour choisir l'algorithme, il faut utiliser le mot clé global ALGO_RESO_CONT.
La méthode du point fixe (ALGO_RESO_CONT='POINT_FIXE') est la plus robuste mais aussi la plus coûteuse puisque le problème non-linéaire (plasticité par exemple) est résolu à chaque changement des statuts de contact.
La méthode de Newton généralisée (ALGO_RESO_CONT='NEWTON') est plus performante mais pose parfois des problèmes de convergence en dynamique. Dans ce cas on reviendra vers une méthode de point fixe.

3.3 Résolution d'un problème avec frottement

3.3.1 Traitement de la non-linéarité de seuil

Dans *Code_Aster*, le seul modèle de frottement disponible est celui de Coulomb (cf. [R5.03.50]). Une non-linéarité supplémentaire doit être traitée en présence de frottement : c'est la non-linéarité de seuil.
Le seuil de frottement dépend en effet de la pression de contact qui est elle-même inconnue.
La loi de Coulomb fait intervenir un coefficient μ , appelé coefficient de Coulomb. Pendant la phase dite d'adhérence, un point en contact ne bouge pas (il a une vitesse nulle et il existe une réaction tangentielle). Pendant la phase de glissement, le point a une vitesse non nulle et est soumis à une réaction tangentielle égale à μ fois la réaction normale.

En général, **si le coefficient de frottement est très faible, il est conseillé de négliger les frottements**. Par ailleurs, il est conseillé dans les études de ne traiter **dans un premier temps que le contact**, ceci afin d'introduire les non-linéarités les unes après les autres.

Les méthodes discrètes qu'elles travaillent par pénalisation ou dualisation s'appuient sur des algorithmes dédiés en présence de frottement (distincts de ceux utilisés pour le contact) tandis que la méthode continue utilise deux algorithmes différents :

- méthode de point fixe sur les seuils de frottement : le seuil est actualisé dans une boucle externe à la boucle de Newton (et à la boucle sur les statuts de contact) ;
- méthode de Newton généralisée : la non-linéarité de frottement est traitée dans le processus de Newton, par dérivation explicite de tous les termes non-linéaires.

3.3.2 Formulation discrète : dualisation du frottement (algorithme 'LAGRANGIEN')

Ce type d'algorithme est bien adapté au traitement de modèles 2D, il ne nécessite **aucun** paramétrage et converge relativement bien.

Pour des problèmes 3D, la convergence se révèle plus difficile en particulier dès que le coefficient de frottement devient plus grand que 0,1. Si on observe que la convergence est très lente mais que le résidu décroît, il est possible d'accélérer le calcul en renseignant le coefficient COEF_MATR_FROT à une valeur de 0,5 (*cf.* [U4.44.11] pour la signification de ce paramètre).

3.3.3 Formulation discrète : pénalisation du frottement (algorithme 'PENALISATION')

Pour les problèmes 3D ou bien de grande taille, il est conseillé de traiter le problème de frottement par pénalisation. Cela demande, comme pour la pénalisation du contact, l'entrée d'un paramètre de pénalisation (E_T). Plus difficile à choisir que son équivalent E_N, il nécessite de réaliser une petite étude paramétrique.

Pour faire l'analogie avec le cas de la pénalisation du contact on remarquera que la phase d'adhérence à proprement parler disparaît (dès que le contact est activé il y a interpénétration, en frottement il y a toujours glissement).

La convergence peut aussi être accélérée par l'utilisation du mot-clé COEF_MATR_FROT.

3.3.4 Formulation 'CONTINUE'

C'est la méthode de choix lorsque l'on doit traiter un problème de contact-frottement : elle est la plus robuste de plus elle tolère bien les grands coefficients de frottement (plus grands que $0{,}3$).

Il est possible de choisir parmi deux algorithmes de résolution pour le frottement avec le mot-clé ALGO_RESO_FROT.
La méthode du point fixe (ALGO_RESO_FROT='POINT_FIXE') est robuste mais coûteuse puisque le problème non-linéaire (plasticité par exemple) est résolu à chaque changement du seuil de frottement et à chaque changement des statuts de contact.
La méthode de Newton généralisée (ALGO_RESO_FROT='NEWTON', choix par défaut) est très performante et offre un bon niveau de robustesse.
Le gros avantage de cet algorithme est sa moindre dépendance à la valeur du coefficient de frottement, puisqu'il n'y pas de boucle sur les seuils. On produit une matrice tangente non-symétrique, ce qui représente un léger surcoût lors de la factorisation et limite la gamme des solveurs itératifs utilisables.
Il est préférable d'utiliser la méthode de Newton généralisée dès lors que le coefficient de frottement n'est pas négligeable. Les gains en temps calcul sont très importants (jusqu'à 80% de gain par rapport au point fixe).
Les deux algorithmes donnent des résultats identiques.

Lorsque toutefois des difficultés de convergence apparaissent, notamment en présence de glissements importants, l'utilisateur pourra paramétrer le coefficient COEF_FROT (qui a la dimension de l'inverse d'une distance). Ce paramètre prend une valeur de 100 par défaut : on essaiera des valeurs comprises entre 10^{-6} et 10^{6} . Pour des études où l'adhérence est prépondérante, on favorisera des valeurs de COEF_FROT inférieures à la valeur par défaut tandis que pour des cas où le glissement est prépondérant, on choisira des valeurs supérieures.

3.4 Récapitulatif pour le choix des méthodes de résolution

3.4.1 Pour le contact-frottement

Pour les problèmes à faible nombre de degrés de liberté en contact (inférieur à 500 degrés de liberté), on privilégiera une formulation discrète avec algorithme des contraintes actives ('CONTRAINTE'). Si le frottement doit être activé, on se tournera vers une formulation 'CONTINUE'.

Pour les problèmes à grand nombre de degrés de liberté en contact (supérieur à 500 degrés de liberté), l'algorithme de résolution itératif par contraintes actives 'GCP' est le plus approprié. Si toutefois on doit prendre en compte du frottement, on pourra se tourner une nouvelle fois vers la formulation 'CONTINUE'.

Pour les problèmes de grande taille (indépendamment du nombre de degrés de liberté en contact), la résolution du système linéaire consomme une grande partie du temps de calcul, le choix du solveur linéaire est donc primordial. La méthode 'CONTINUE' (et dans une moindre mesure la méthode par 'PENALISATION') est bien adaptée en ce sens qu'elle laisse à l'utilisateur le choix du solveur linéaire et qu'elle est bien parallélisée.

3.4.2 Pour le système linéaire

Si on utilise une formulation discrète (hors pénalisation), seuls les solveurs linéaires directs sont accessibles. On choisira donc le solveur 'MULT_FRONT' sauf si on réalise un calcul parallèle auquel cas on sélectionnera 'MUMPS'. La méthode 'GCP' alliée au solveur linéaire 'MUMPS' profite d'un bon niveau de parallélisation dans l'algorithme de contact.

Si on utilise une formulation continue, il est conseillé, dès que le problème global dépasse 100 000 degrés de liberté, d'utiliser un solveur itératif associé au pré-conditionneur 'LDLT_SP' et à la méthode de Newton-Krylov (cf. § 3.2.3). Si le calcul met en œuvre du frottement ou bien est parallèle, le solveur itératif 'PETSC' est le meilleur choix.

4 Méthodologies

Dans cette partie, on répond aux questions fréquemment posées lors des études avec du contact-frottement. Les techniques mises en place dans cette partie s'appuient souvent sur d'autres opérateurs que `DEFI_CONTACT`, on décrira brièvement les mot-clés à utiliser mais l'utilisateur pourra avantageusement se reporter aux documentations d'utilisation de ces commandes.

4.1 Récupérer la pression de contact

En post-traitement d'un calcul de contact, on souhaite généralement accéder aux efforts de contact-frottement. Plus précisément, on souhaite connaître la contrainte normale et tangentielle sur le bord des solides en contact.
La formulation continue du contact donne directement accès à la pression de contact-frottement, tandis que les formulations discrètes nécessitent de l'approximer par les contraintes sur le bord.
Un exemple de mise en œuvre pour les deux formulations existe dans le cas-test SSNP154 [V6.03.154].

4.1.1 Formulation continue

En formulation continue, le champ `DEPL` contient une ou plusieurs inconnues supplémentaires :

- `LAGS_C` représente la densité surfacique d'effort de contact exprimée sur la configuration de référence.
- `LAGS_F1` et `LAGS_F2` représentent les coordonnées d'un vecteur directeur dans le plan tangent. Ce vecteur de norme inférieure ou égale à 1 indique la direction de glissement ou d'adhérence lorsque que l'on prend en compte le frottement.

Ces grandeurs sont définies en tout point de la surface esclave de contact. On peut donc facilement accéder à la pression de contact. On notera cependant qu'en grands déplacements, configuration initiale et finale n'étant plus confondues, le degré de liberté `LAGS_C` n'a plus la signification d'une pression.

Pour accéder à la densité surfacique d'effort de frottement (en phase d'adhérence comme de glissement), il faut réaliser un calcul supplémentaire : la norme du vecteur directeur dans le plan tangent donne en effet l'amplitude par rapport au seuil de frottement.
Si on note λ la pression de contact alors la densité d'effort de frottement τ s'écrit :

$$\tau=\mu.\lambda.\sqrt{\left(LAGS_{F1}^2+LAGS_{F2}^2\right)} \quad (4)$$

En formulation pénalisée (`ALGO_CONT='PENALISATION'`), les degrés de liberté de pression continuent d'exister, on peut donc appliquer ce qui précède.

Il arrive parfois que la pression de contact relevée par cette méthode présente des oscillations, en particulier pour des géométries courbes. Dans ce cas, les résultats obtenus restent justes mais il devient difficile d'exploiter la pression de contact directement. On renvoie alors au paragraphe suivant pour le relevé de la pression de contact par les contraintes.

4.1.2 Formulation discrète

En formulation discrète, aucun degré de liberté n'est rajouté aux inconnues principales. Le problème de contact étant formulé sur le système discret, les éventuels multiplicateurs de Lagrange utilisés n'ont même pas la dimension d'une pression mais celle de forces nodales.
Cette absence oblige à calculer le tenseur des contraintes de Cauchy sur le bord des surfaces en contact. La pression de contact s'écrit en effet :

$$\lambda=(\boldsymbol{\sigma}.\underline{n}).\underline{n} \quad (5)$$

où $\underline{n}$ est la normale à la surface de contact et σ le tenseur des contraintes de Cauchy.

Pour calculer les contraintes de Cauchy sur le bord, il faut les interpoler à partir des contraintes aux points de Gauss. Pour ensuite obtenir la pression, il faut calculer les normales sur le bord (de la géométrie déformée en grands déplacements) puis créer un champ à partir des contraintes interpolées et des normales.
L'encadré ci-dessous montre comment on peut procéder pour un calcul 2D. Cela suppose d'avoir déjà calculé dans le résultat (RESU) le champ des contraintes de Cauchy aux nœuds (SIEF_NOEU).

```
sigm=CREA_CHAMP(TYPE_CHAM='NOEU_SIEF_R',
                OPERATION='EXTR',
                RESULTAT=RESU,
                NOM_CHAM='SIEF_NOEU',
                INST=1.0,
                );

depl=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                OPERATION='EXTR',
                RESULTAT=RESU,
                NOM_CHAM='DEPL',
                INST=1.0,
                ) ;

# normale sur la configuration initiale
NormaleI=CREA_CHAMP(TYPE_CHAM='NOEU_GEOM_R',
                   OPERATION='NORMALE',
                   MODELE=MO,
                   GROUP_MA=('Slave','Master',),
                   ) ;

Pression = FORMULE(VALE='SIXX*X*X+SIYY*Y*Y+2*SIXY*X*Y',
                   NOM_PARA=('SIXX','SIYY','SIXY','X','Y',),);

Pres=CREA_CHAMP(TYPE_CHAM='NOEU_NEUT_F',
                OPERATION='AFFE',
                MAILLAGE=MESH,
                AFFE=_F(GROUP_MA=('Slave','Master',),
                        NOM_CMP='X1',
                        VALE_F=Pression,),);

pI=CREA_CHAMP(TYPE_CHAM='NOEU_NEUT_R',
             OPERATION='EVAL',
             CHAM_F=Pres,
             CHAM_PARA=(NormaleI,sigm,),);
```

Pour un calcul en grands déplacements, la normale doit être calculée sur la configuration déformée. Pour cela il faut translater le maillage avec la commande MODI_MAILLAGE.
Dans l'exemple ci-dessus, on utilise la formule tirée de (5) pour calculer explicitement la pression de contact. Dans le cas particulier où le bord sur lequel on extrait la pression est parallèle aux axes du repère, la pression est directement égale à une des composantes diagonales du tenseur des contraintes de Cauchy (SIXX, SIYY ou SIZZ).

4.2 Mouvements de corps rigide bloqués par le contact

Ce paragraphe ne s'applique qu'aux études en statique. En dynamique, les mouvements de corps rigide sont permis.

Il arrive dans les études que le contact permette de bloquer les mouvements de corps rigide de certains solides (et faire en sorte que ceux-ci se déforment). La non-prise en compte initiale de ce phénomène entraînera donc la singularité de la matrice de rigidité (et donc l'impossibilité de résoudre).

Les formulations discrètes ne sont pas adaptées à une prise en compte initiale du contact, la réalisation d'études avec des solides uniquement tenus par le contact va donc nécessiter dans ce cas un enrichissement de la modélisation.
La formulation continue permet de tenir naturellement compte d'un contact initial et à ce titre est donc bien adaptée à l'étude de mécanismes.

Pour des études en trois dimensions, il existe 6 mouvements de corps rigide possibles : 3 translations, 3 rotations. Pour des études en deux dimensions (modélisations D_PLAN, C_PLAN), il existe 3 mouvements de corps rigide : 2 translations et une rotation. La modélisation axisymétrique (AXIS) est particulière : il n'existe qu'un seul mouvement de corps rigide, la translation suivant l'axe Oy (axe de la symétrie cylindrique).

Lorsque l'on constate l'existence de mouvements de corps rigide dans sa modélisation, **on commencera toujours par vérifier qu'il n'existe pas des symétries dans la structure et son chargement**. Les conditions de symétrie permettent en effet de supprimer une grande partie des mouvements de corps rigide.

Un exemple de blocage de mouvements de corps rigide en formulation continue (par CONTACT_INIT) et en formulation discrète (par des ressorts) est disponible dans le cas-test SSNA122 [V6.01.122].

4.2.1 Formulation continue

En formulation continue, la prise en compte d'un contact initial est assurée zone par zone avec le mot-clé CONTACT_INIT. Par défaut au début d'un calcul toutes les liaisons à jeu nul (ou interpénétrées) sont activées (CONTACT_INIT='INTERPENETRE'). La tolérance, pour déterminer si un jeu est nul ou interpénétré, est fixée en interne dans le programme à $10^{-6}\times a_{min}$ où a_{min} représente la plus petite arête non nulle du maillage.

Il est possible de désactiver cette activation automatique (CONTACT_INIT='NON'). Lorsque l'on fait des calculs non-linéaires avec reprise (c'est-à-dire avec le mot-clé ETAT_INIT de STAT_NON_LINE), il est indispensable d'utiliser la valeur par défaut ('INTERPENETRE') afin d'assurer une reprise à partir du vrai état de contact (et non pas d'un état vierge).

Enfin si l'on veut coller initialement toutes les surfaces de contact indépendamment du jeu initial, on pourra sélectionner CONTACT_INIT='OUI' (cela peut être utile si les maillages ne sont pas parfaitement en contact).

Dans tous les cas de figure où un contact initial est déclaré, des efforts seront générés : **il ne s'agit pas d'un simple repositionnement géométrique** visant à coller les maillages.

L'activation d'un contact initial bloque les mouvements de corps rigide suivant la direction normale à la surface. Si l'on veut prendre en compte un état adhérent initial afin de bloquer la direction tangente, on pourra spécifier un seuil initial de contact non nul *via* SEUIL_INIT. Ce paramètre renseigne la valeur initiale de la pression de contact (homogène à une densité de force surfacique).

Il convient de noter que l'utilisation d'un contact initial en formulation continue permet aussi de s'affranchir de la non-convergence lorsque qu'une structure n'est soumise qu'à des déplacements. Par exemple, lorsque deux solides initialement en contact sont pressés l'un contre l'autre par des déplacements (il s'agit donc d'un mouvement de corps rigide).

4.2.2 Formulation discrète

En formulation discrète, on est obligé de bloquer manuellement les mouvements de corps rigide du solide incriminé par des ressorts de faible raideur. Par « faible » on entend suffisamment petite pour ne générer que des forces nodales négligeables devant les forces nodales mises en jeu dans le calcul.
Le but des ressorts est de faire en sorte que le calcul sans contact soit capable de tourner en mécanique linéaire (c'est-à-dire dans l'opérateur MECA_STATIQUE ou bien dans STAT_NON_LINE une fois retiré les conditions de contact).

Il existe deux approches pour l'ajout de ressorts :
- rajouter un ressort de faible raideur en tout point de la structure
- ajouter des ressorts en des points bien choisis pour bloquer les mouvements de corps rigide de la structure

La première approche a l'avantage de la généricité mais peut parfois trop perturber la solution (quelle que soit la raideur des ressorts). En effet une telle approche revient à ajouter sur tous les termes diagonaux de la matrice un terme positif qui la rend inversible.
La seconde approche ne rajoute des ressorts que là où c'est nécessaire. Lorsqu'il existe des points de la structure qui seront amenés à avoir un déplacement faible (donc à ne générer qu'une force nodale faible dans le ressort), cette approche est plus adaptée.

Pour appliquer un ressort dans *Code_Aster*, il faut créer des mailles de type `POI1` à partir de nœuds. Pour cela on utilise l'opérateur `CREA_MAILLAGE/CREA_POI1`. Pour utiliser la première approche on choisira de créer ce groupe de mailles sur toute la structure (`TOUT='OUI'`), tandis que pour la seconde approche, on indiquera le groupe de nœuds souhaité. Le groupe de mailles nouvellement créé servira à affecter une modélisation de type `'DIS_T'` ou `'2D_DIS_T'` dans `AFFE_MODELE`.

La définition des caractéristiques du ressort s'effectue dans l'opérateur `AFFE_CARA_ELEM`. Par défaut, les raideurs sont entrées dans le repère global. Si par exemple on souhaite bloquer un mouvement de corps rigide dans une direction parallèle aux axes du repère global, on ne définira une raideur non nulle que suivant cette direction. Ci-dessous un exemple de définition d'une raideur pour un calcul 2D suivant la direction `DY`.

```
RESSORT=AFFE_CARA_ELEM(MODELE=model,
                       DISCRET_2D=_F(CARA='K_T_D_N',
                                     GROUP_MA='SPRING',
                                     VALE=(0.,1.0e-1,),),);
```

Dans les cas où la direction à bloquer n'est pas parallèle aux axes, deux alternatives sont possibles :
- définir une raideur suivant toutes les directions
- définir la raideur dans un repère local. Il faut alors définir l'orientation de ce repère (mot-clé `ORIENTATION` de `AFFE_CARA_ELEM`) ou bien utiliser des ressorts s'appuyant non plus sur des mailles `POI1` mais `SEG2`.

Pour un exemple d'utilisation de ressorts, on consultera le cas-test ZZZZ237 et sa documentation [V1.01.237].

4.3 Grandes déformations, grands déplacements et contact

La prise en compte de conditions de contact-frottement est totalement découplée de la prise en compte de grands déplacements ou de grandes déformations. Plus généralement toute non-linéarité quelle soit d'ordre matériau ou géométrique est *a priori* compatible avec l'utilisation du contact.

En pratique, on constate souvent des difficultés de convergence dans des études mêlant les trois non-linéarités. On donne dans la suite de cette section la démarche à adopter dans ce cas.

Des exemples de calcul mélangeant les trois non-linéarités sont disponibles dans les cas-tests SSNP155 [V6.03.155], SSNP157 [V6.03.157] et SDNV103 [V5.03.103].

4.3.1 Découpler les non-linéarités

Lorsque qu'un tel calcul échoue, la première démarche est de revenir en arrière : en découplant les non-linéarités et en essayant d'appliquer les bonnes pratiques en non-linéaire (*cf.* [U2.04.01]).
Cela signifie :
- réaliser un calcul élastique en petites perturbations avec le contact activé. Si ce calcul échoue, appliquez les conseils livrés dans la première partie de ce document (orientation des normales, choix des surfaces maîtres et esclaves, choix de l'algorithme de résolution, …)
- réaliser un calcul avec une loi de comportement non linéaire mais sans contact. Si celui-ci échoue, alors le problème vient de l'intégration du comportement. On se rapportera alors aux documentations [U2.04.02] et [U2.04.03].
- si nécessaire réaliser un calcul en grands déplacements mais sans contact et sans non-linéarité matériau. Si ce calcul ne fonctionne pas, essayez d'utiliser un autre modèle parmi ceux de

grands déplacements disponibles dans *Code_Aster* ('SIMO_MIEHE', 'GDEF_LOG', 'PETIT_REAC').

4.3.2 Bien paramétrer l'algorithme de Newton

Si le calcul complet (mêlant toutes les non-linéarités) ne converge pas malgré l'application des conseils précédents alors on peut essayer de jouer sur les paramètres de l'algorithme de Newton. Cela part du constat suivant :
Lorsque l'on couple contact et non-linéarité matériau par exemple, il est possible (par la correction « brusque » du contact) de déclencher dans la loi de comportement des mécanismes (sortie du domaine élastique, décharge) qui ne devraient pas être actifs dans la solution finale et qui risquent de dégrader la matrice tangente (jusqu'à la rendre non inversible). Cela rend alors toute convergence impossible.

On propose donc d'utiliser les réglages suivants dans l'algorithme de Newton (opérateur STAT_NON_LINE ou DYNA_NON_LINE) :

- réactualisation de la matrice tangente à chaque itération (REAC_ITER=1)
- utilisation d'une prédiction élastique (PREDICTION='ELASTIQUE')
- en grandes déformations (DEFORMATION='SIMO_MIEHE'), la matrice tangente est non symétrique, il faut donc bien veiller à renseigner SYME='NON' dans le mot-clé SOLVEUR.

Lorsque le calcul a encore du mal à converger, il faut revenir à la modélisation :

- est-ce que mon calcul provoque des problèmes d'incompressibilité ? Dans ce cas, consultez les documentations [U2.04.01][U2.04.02] et essayez d'utiliser des éléments finis adaptés (sous-intégrés, à formulation mixte).
- est-ce que le comportement que j'utilise dispose d'une matrice tangente cohérente ? Si ce n'est pas le cas, on peut essayer en dernier recours d'utiliser une matrice 'ELASTIQUE' et augmenter le nombre d'itérations de Newton.

4.3.3 Résolution d'un problème quasi-statique en dynamique lente

En dernier recours, pour les problèmes quasi-statiques, réaliser un calcul dynamique en temps long peut apporter une solution. La matrice de masse a pour effet de stabiliser la structure, il faut toutefois s'assurer que les forces d'inertie restent alors faibles devant les forces internes du système.

On conseille pour ce type de modélisation d'affecter à la structure sa vraie masse volumique (c'est de toute manière obligatoire en présence de chargement de pesanteur) et de réaliser le calcul en utilisant de grands pas de temps.
Un exemple de mise en œuvre est disponible dans le cas-test SSNP155 [V6.03.155].

4.4 Surface rigide et contact

On souhaite parfois modéliser dans les études des solides rigides qui entrent en contact avec des solides déformables. Dans cette section, on explique comment optimiser de telles études.

Afin de ne pas alourdir la modélisation les solides rigides ne seront pas modélisés entièrement : **seul leur bord portera des degrés de liberté**. Afin de faciliter l'orientation des normales de ce solide rigide le maillage comportera cependant le solide complet.

Après avoir orienté les normales, on affectera donc dans AFFE_MODELE uniquement des éléments de bord à la peau du solide rigide : comme les éléments de bord ne portent pas de rigidité, une alarme est émise pour prévenir du risque de matrice de rigidité non inversible. **Cette alarme est normale dans ce cas** et peut être ignorée.

Pour empêcher que la matrice de rigidité soit singulière, **il faut imposer le déplacement de tous les degrés de liberté portés par le bord rigide**. Cela est fait avec les commandes :

- AFFE_CHAR_CINE/MECA_IMPO dont l'avantage est d'éliminer les inconnues
- AFFE_CHAR_MECA/DDL_IMPO qui ajoute des inconnues supplémentaires au problème.

On conseille donc d'éliminer les inconnues (AFFE_CHAR_CINE).
La surface rigide sera déclarée comme **surface maître** dans DEFI_CONTACT comme expliqué au §2.2.1.

On pourra se reporter au cas-test SSNV506 [V6.04.506] pour un exemple de contact avec surface rigide.

4.5 Redondance entre conditions de contact-frottement et conditions aux limites (symétrie)

En présence de symétries dans la structure étudiée, il est courant que les conditions de frottement entrent en conflit avec les conditions aux limites de symétrie. La Figure 4.5-1 montre l'exemple de deux cubes en contact-frottement, la partie hachurée représente les faces des cubes soumis à une condition de symétrie (DX=0).
Dans cet exemple, l'arête du cube supérieur **en trait épais** appartient à la surface esclave et porte également la condition de symétrie. Cette condition entre en conflit avec la condition de frottement écrite dans le plan tangent (ici le plan xOz). En pratique le calcul s'arrêtera une fois le contact établi car la matrice tangente sera singulière.
Mécaniquement on voit que la condition de symétrie implique que l'adhérence ou bien le glissement ne se produira que suivant la direction DZ (vecteur tangent vert). Pour éliminer la redondance il faut donc exclure la direction de frottement suivant DX (vecteur tangent rouge).

Pour cela on utilisera le mot-clé SANS_GROUP_NO_FR pour désigner la liste de nœuds de l'arête esclave puis on renseignera (dans le repère global) DIRE_EXCL_FROT=(1,0,0) soit la direction DX à exclure.

Le cas-test ZZZZ292 met en œuvre la fonctionnalité SANS_GROUP_NO_FR.

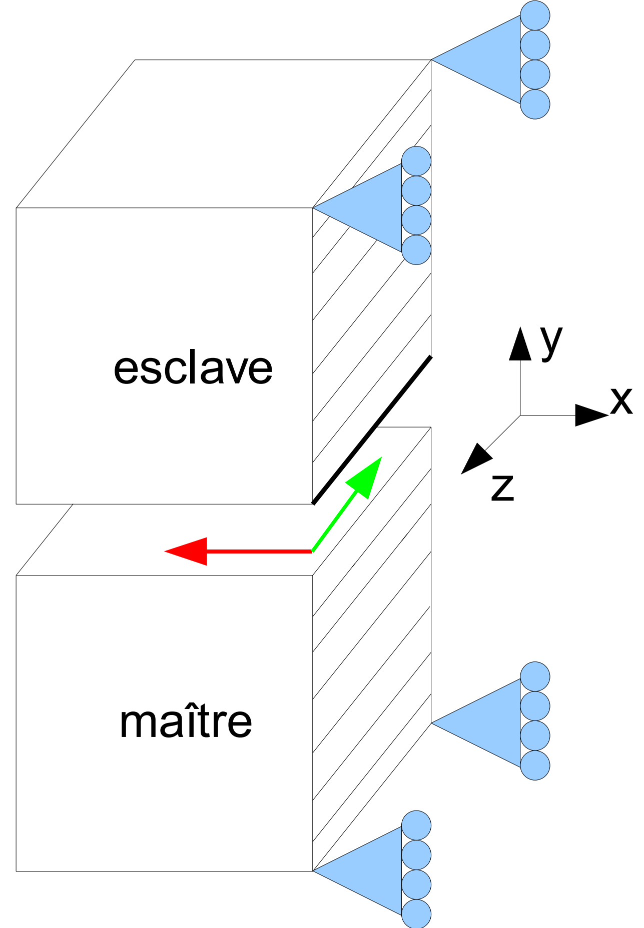


Figure 4.5-1: Élimination de directions de frottement

4.6 Mesurer l'interpénétration sans résoudre le contact

La résolution d'un problème de contact pouvant parfois coûter cher il peut être avantageux de remplacer l'imposition des conditions de contact par une simple vérification de l'interpénétration. C'est d'autant plus intéressant dans les cas où l'on souhaite simplement vérifier que des solides ne vont pas entrer en contact.
Pour chacune des zones de contact définies dans l'opérateur `DEFI_CONTACT`, il est possible de choisir si on souhaite y faire respecter le contact (`RESOLUTION='OUI'`) ou non (`RESOLUTION='NON'`).

L'intérêt d'une telle approche est de ne pas alourdir un calcul : lorsque qu'un calcul mené sans résolution **sur l'intégralité des zones de contact** montre qu'il n'y a pas d'interpénétration alors on peut ignorer la modélisation du contact.
Attention cependant : si au moins une des zones de contact est « résolue » et une autre « non résolue » alors l'existence d'une interpénétration ne préjuge pas de la solution d'un calcul complet avec contact (à cause des interactions possibles entre zones de contact).

Enfin cette technique peut aussi servir à mesurer le taux d'interpénétration au niveau des zones de contact pour calibrer un critère comme le coefficient de pénalisation ou l'interpénétration maximum tolérée dans la méthode de résolution `'GCP'` .

4.7 Visualiser les résultats d'un calcul de contact

Lorsque l'on visualise les résultats d'un calcul de contact-frottement dans un logiciel de post-traitement, il faut prendre garde à plusieurs choses :

- pour l'affichage des déformées, **un facteur d'amplification différent de 1 peut conduire à visualiser des interpénétrations non réelles**
- pour les calculs 2D en formulation 'CONTINUE', on fera attention lors de l'affichage de déformées aux logiciels de post-traitement qui considèrent les trois premières composantes d'un champ comme les composantes suivant X, Y et Z du déplacement. En 2D, la troisième composante correspond à LAGS_C et doit donc être ignorée
- lors de la visualisation du champ de post-traitement du contact (VALE_CONT) et plus particulièrement de la composante CONT qui indique l'état du contact, on fera attention à l'interpolation des champs aux nœuds réalisée parfois automatiquement. En effet cette composante prend les valeurs 0 (pas de contact), 1 (contact adhérent) ou 2 (contact glissant). L'état adhérent n'est possible qu'en présence de frottement : si on visualise une telle valeur pour un calcul de contact sans frottement c'est qu'il y a interpolation du champ.

4.8 Contact ponctuel avec des éléments discrets (ressorts)

Les éléments discrets (ou ressorts) 2D_DIS_T* ou DIS_T* associés à la loi de comportement DIS_CHOC [R5.03.17] permettent de rendre compte d'un contact ponctuel dans une direction fixée. Ils sont bien adaptés à la modélisation de chocs et sont à ce titre souvent utilisés en dynamique sur base modale [U4.53.21] et en dynamique explicite [U4.53.01].
Les ressorts peuvent s'appuyer indifféremment sur une maille ponctuelle ou un segment. Dans tous les cas, il est nécessaire d'orienter correctement chaque élément avec la commande AFFE_CARA_ELEM [U4.42.01].
Le contact comme le frottement sont résolus par pénalisation (cf. §3.2.2). Les raideurs de pénalisation, le coefficient de frottement ainsi que les jeux initiaux sont précisés dans le matériau DIS_CONTACT (commande DEFI_MATERIAU, [U4.43.01]).

Ce type d'élément n'est pas utilisable en grands déplacements car la direction de contact est fixe et donnée par l'orientation initiale de l'élément discret.

Les cas-tests SSNL130A et SDND100C mettent en œuvre des ressorts de contact.

4.9 Éléments de joints (hydro-)mécaniques avec contact et frottement

Les éléments de joints (hydro-)mécaniques PLAN_JOINT(_HYME) et 3D_JOINT(_HYME) permettent de modéliser l'ouverture d'une fissure sous la pression d'un fluide et le frottement sur les bords de la fissure fermée avec la loi JOINT_MECA_FROT [R7.01.25]. Il est possible de coupler l'ouverture de la fissure et la propagation du fluide avec les modélisations *_HYME.
La formulation du contact-frottement est pénalisée et les paramètres afférents sont renseignés sous le mot-clé JOINT_MECA_FROT de la commande DEFI_MATERIAU [U4.43.01].

Les cas-tests SSNP142C et SSNP142D fournissent un exemple d'application de tels éléments sur la modélisation d'un barrage.

5 Bibliographie

[1] M. ABBAS. “Formulation discrète du contact-frottement”. Documentation de référence de *Code_Aster*. [R5.03.50].

[2] M. ABBAS. “Éléments de contact dérivés d'une formulation hybride continue”. Documentation de référence de *Code_Aster*. [R5.03.52].

[3] T. DE SOZA. « Opérateur DEFI_CONTACT ». Documentation d'utilisation de *Code_Aster*. [U4.44.11].

[4] A. KONTER. « Advanced Finite Element Contact Benchmarks ». NAFEMS, 2006.

[5] A. KONTER. « How to undertake a contact and friction analysis ». NAFEMS, 2000.